Project 1

Group 06: Fredrick Nilsson

November 20, 2023

Exercise 1

Assume that we can perform ten million tests of the above kind each second on our computer. How long would it take to factor a 25 digit number (with two prime factors both of the same order (12 digit numbers))?

If we assume that the 25-digit number is the largest posssible, 9,999,999,999,999,999,999,999,999, and that both of the two prime factors are the smallest possible 12-digit prime number, 100,000,000,003.

$$9999999999999999999999999/(100000000003^2) \approx 1000$$

Then the remaining prime factors have has to be smaller than 1000. So we can assume that these factors will be discovered in the first 0.0001 seconds of the program.
This only leaves the two 12-digit primes to be discovered. Which will take quite a bit more time.
But when the first 12-digit prime factor is found, we can easily figure out the last factor by dividing the 25-digit number by the known factors, therefore we only need to find one of them.
As the smallest 12-digit prime number is 100 000 000 003 it will in the best case take

$$100000000003/10000000 = 1000.00000003$$

seconds for it get solved. Since the largest 12-digit prime number is 999 999 999 961 it will in the worst case take

$$999999999961/10000000 = 9999.99999961$$

seconds for it get solved.
If we assume that the rest of the 12-digit primes have a close to even distribution between the best and worst case, we can assume that the average time it takes to find a 12-digit prime factor is

$$(1000.00000003 + 9999.99999961)/2 = 5499.99999982$$

seconds.

Exercise 2

If you want to factor many 25 digit numbers, you can improve the running time of the basic trial division by first precomputing and storing the primes up to

$$\sqrt{N}$$

How much faster does your improved trial division algorithm become for 25 digit numbers? Roughly how much storage does your algorithm require? What kind of budget does the storage requirement demand; student budget, big government grant, more dollars than there are atoms in the universe? You may check current storage pricing on http://www.prisjakt.nu.

If we assume that the 25-digit number is the largest posssible, 9,999,999,999,999,999,999,999,999, then the square root of that number is approximately 3,162,277,660,168. So we need to precompute and store all the primes up to 3,162,277,660,168.
According to the prime number theory[1], the amount of primes smaller than 3,162,277,660,168 can be approximated as

$$\frac{3,162,277,660,168}{ln(3,162,277,660,168)} \approx 109,868,779,044$$

.
As the largest prime that could possibly be stored is smaller than 3,162,277,660,168, we can assume that no number prime will need more than 42 bits to store, since 3,162,277,660,168 = 101110000001000110011011111100011000001000 in binary. If we then assume that all the primes will be stored in 42 bits each, the total size to store all the primes is

$$109,868,779,044 * 42 = 4,614,488,719,848$$

bits. Which is approximately 577 gigabytes. Although this is assuming a close to optimal way to store the primes, so we can assume that the actual storage size will be a bit larger.
According to prisjakt.nu, we can buy an 1TB Seagate Barracuda ST1000DM010 harddrive for 559 sek. So we can assume that the storage requirement can be satisfied by a student budget.

Exercise 3

Factoring my number, 127423368713324519534591, I found the factors 312709043917 and 407482198523.

It took 15544 ms (15s 544ms) for the program to find them on a M1 Macbook Air (2021).

Exercise 4

Factoring the number 92434447339770015548544881401 resulted in the factors 727123456789451 and 127123456789451.

It took 401126 ms (6m 41s 126ms) for the program to find them on a M1 Macbook Air (2021).

Program printout

```
[2023-11-20T14:41:40.151Z INFO  project1::exercise4] Let's find some factors of N=92434447339770015548544881401
[2023-11-20T14:41:40.151Z INFO  project1::exercise4] Got bound B = 10000
[2023-11-20T14:41:40.151Z INFO  project1::exercise4] Sieving up to 10000
[2023-11-20T14:41:40.225Z INFO  project1::exercise4] Waiting for threads to finish, please wait
[2023-11-20T14:48:21.240Z INFO  project1::exercise4] Might have found something:
    1, 92434447339770015548544881401 => n = 92434447339770015548544881401*1 = 92434447339770015548544881401
[2023-11-20T14:48:21.261Z INFO  project1::exercise4] Might have found something:
    127123456789451, 727123456789451 => n = 727123456789451*127123456789451 = 92434447339770015548544881401
[2023-11-20T14:48:21.261Z INFO  project1::exercise4] Two factors of 92434447339770015548544881401 are
    (727123456789451, 127123456789451)
Took 401126ms
```

Working hours

I used approximately 1 hour for exercise 1 and 2. But exercise 3 and 4 took approximately 20 hours as I had to rewrite the program several times.

References

[1] Wikipedia. Prime number theorem. https://en.wikipedia.org/wiki/Prime_number_theorem, 2023. [Online; accessed 2-November-2023].